N. B. Meisner *et al.*, "Time division digital bus techniques implemented on coaxial cable," in *Proc. Computer Networking Symp.* (National Bureau of Standards, Gaithersburg, MD, Dec. 15, 1977).
R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kurnzelman, "Advances in packet radio technology," this issue, pp. 1468–1496.
P. Mockapetris *et al.*, "On the design of local network interfaces," *Informat. Process.*, vol. 77, pp. 427–430, Aug. 1977.
ARPANET Protocol Handbook, Network Information Center, SRI International, Menlo Park, CA, NIC 7014, revised Jan. 1978.
V. Cerf and R. Kahn, "A protocol for packet network interconnector," *IEEE Trans. Commun.*, vol. COM-25, No. 1, pp. 169–178, May 1974.
L. Pouzin, "Virtual circuits vs. datagrams–Technical and political problems," in *AFIPS Conf. Proc.* (National Computer Conf., June 1976), p. 483.
D. H. Crocker *et al.*, "Standard for the format of ARPA network text messages," ARPA Network RFC 733, NIC 41952, Nov. 21, 1977.
[30] R. H. Thomas, "A resource sharing executive for the ARPANET," *AFIPS Conf. Proc.*, vol. 42 (Nat. Computer Conf. and Exposition, 1973), pp. 155–163.
[31] D. J. Farber and F. H. Heinrich, "The structure of a distributed computer system–The distributed file system," in *Proc. Int. Conf. on Computer Communication* (Washington, DC, 1972), pp. 364–370.
[32] E. G. Manning and R. W. Peebles, "A homogeneous network for data sharing communications," Computer Communications Network Group, University of Waterloo, Waterloo, ON, Tech. Rep. CCNG-E-12, Mar. 1974.
[33] V. G. Cerf and P. T. Kirstein, "Issues in packet network interconnection," this issue, pp. 1386–1408.
[34] S. L. Ratliff, "A dynamic routing algorithm for a local packet network," S.B. thesis, M.I.T., Department of Electrical Engineering and Computer Science, Cambridge, MA, Feb. 1978.

Enhanced Message Addressing Capabilities for Computer Networks

JOHN M. McQUILLAN, MEMBER, IEEE

Invited Paper

Abstract—Three message addressing modes are described:

1) Logical addressing, in which a permanently assigned address denotes one or more physical addresses. This permits multiple connections of the subscriber to the network, as well as other functions.

2) Broadcast addressing, in which a message is addressed to all subscribers.

3) Group addressing and multidestination addressing, in which a message carries the name of a list of addresses, or the list itself.

These methods facilitate many new ways of using computer networks. The paper focuses on two basic issues for each method: efficiency and reliability, and recommends implementation approaches in each case. Significant performance improvements are possible if these addressing methods are implemented with efficient delivery mechanisms. A distinction is made between virtual circuit and datagram systems; virtual circuits are superior for logical addressing, while datagrams are preferable for broadcast, group, and multidestination addressing.

I. INTRODUCTION

HOW SHOULD one user of a network address messages to other users? The answer to this question is fundamental in defining the appearance of the network to its users. For example, does one user have to know exactly where the other is located, or just the region of the network, or is the address independent of location? Can he identify himself to the network or does the network know who he is automatically? If self-identification is possible, can he have several addresses corresponding to several roles or functions? Can he have multiple connections to the network, and can he move from one location to another without changing his address(es)? Can he send a single message to a group or list of other users (e.g., a mailing list) automatically? Can he set up "conference calls" with other users, and join conferences in progress? Can he send a message to all other users?

These questions are important for several reasons: some addressing modes allow functions which would not be available otherwise (e.g., the ability to send a message to a distribution list without knowing the identity or location of the members of the list), and which are essential for certain types of users and applications. Furthermore, these addressing capabilities offer opportunities for efficient implementations that would not exist otherwise (e.g., a message addressed to a group can be transmitted with fewer packets than the equivalent separately addressed messages). The topic of addressing has received surprisingly little attention to date; the present paper indicates that it may be a fruitful area for further work.

Manuscript received May 15, 1978; revised July 3, 1978.
The author is with Bolt Beranek and Newman, Inc., Cambridge, MA 02138.

0018-9219/78/1100-1517$00.75

The intent of this paper is to identify addressing modes which can be of value, both in providing a useful network interface for users and in permitting efficient network operations within packet-switching networks (though several of our conclusions have broader applicability). We distinguish between the addressing mode, how the user identifies the intended recipient(s) of a message, from the addressing implementation, how the network processes the message. The former is an interface between the user and the network; the latter is a protocol within the network. It is also useful to distinguish between addressing, how the network selects the destination(s) of the message, and routing, how the network selects the path(s) over which the message travels.

We will use the term "node" to refer to the switching computer in the network (the DCE in CCITT terminology), and "subscriber" to refer to the host computer or terminal equipment connected to the network (the DTE). Connection of the subscriber to the node is over an "access line" (which may be a communications circuit) terminating at a "port" on the node. Thus a subscriber can be addressed by the node number and port number to which it is connected; we term this "physical addressing." The address is specified as a part of the "header" attached by the source subscriber to its message. In the basic mode of operation of many computer networks today (see [1], for example), the subscriber presents a message to the network with an address corresponding to the destination subscriber's physical location (see Fig. 1(a)). While this approach is simple and effective, it is also restrictive, since it requires subscribers to know each other's physical locations, and it does not encompass such ideas as assigning a subscriber multiple network connections, or sending a message to more than one address or to one of several addresses, etc.

Our investigations have led us to the conclusion that the following three addressing methods would be valuable additions to most networks:

1) Logical addressing, in which a permanently assigned logical address denotes one or more physical addresses (see Fig. 1(b)). The sender does not need to know the physical location of the destination subscriber, and subscribers can relocate without change of address. Since one logical address can refer to several physical addresses, subscribers can connect to the network by multiple lines ("multiple homing"), increasing reliability and traffic capacity (Fig. 1(c)).

2) Broadcast addressing, in which a message is addressed to all other nodes or subscribers (see Fig. 1(d)). If combined with an efficient implementation, this can reduce network traffic significantly compared with separately addressed messages.

3) Group addressing (Fig. 1(e)) and multidestination addressing (Fig. 1(f)), in which a message carries the name of a list of addresses, or the list itself. When implemented with an appropriate delivery strategy, this also improves performance. It also facilitates electronic mail, conferencing, and similar applications.

This paper describes the considerations involved in the design and implementation of the three methods described above. While there are many issues to be considered, we place the emphasis on efficiency and reliability, since these are the points that lead us to our design choices. The principle of economy of means suggests that an all-purpose addressing mechanism with a single implementation technique would be most desirable. However, we conclude that a different implementation is required in each case to provide the best efficiency levels (to minimize the traffic flowing in the network) and to ensure adequate network reliability (to minimize loss of data due to errors or network failures).

It is appropriate to mention in passing that some networks (e.g., TELENET) have adopted a hierarchical addressing system. Assigning addresses to certain regions of the network, and subaddresses to subscribers within those regions, may result in more compact notation (just as one does not need to dial the area code for a local telephone call) and other operational advantages. Hierarchical addressing can be used in combination with any of the three techniques we discuss in this paper.

These addressing capabilities can be added to many kinds of computer networks; the present paper focuses on the example of packet-switching networks. Virtual circuit networks (in which messages are handled as part of connections analogous to telephone conversations) can support very efficient logical addressing mechanisms because logical addressing information needs to be sent only once per conversation. On the other hand, datagram networks (in which each message is handled independently, like letters in the mail) are less efficient for logical addressing and yet can support broadcast and group addressing more readily because it is unnecessary to set up a complex set of pair-wise conversations. In fact, it may be so unwieldy to install a general multidestination addressing method for virtual circuit service, due to the extensive control required for each circuit, that few virtual circuit networks will offer this service. Of course, addressing is only one of several points of comparison between virtual circuits and datagrams. The debate on the relative merits of the two methods has been continuing for several years (see [2], for example). We hope that this paper contributes some new ideas to that discussion.

II. Logical Addressing and Multiple Homing

A general logical addressing structure can translate many physical addresses into a single logical address and one physical address into many logical addresses. In a virtual circuit network, the logical address is translated by the source node once per connection, permitting all messages in a given virtual circuit to flow to a particular physical address. In a datagram network, on the other hand, the addresses of messages are translated one by one and messages can flow to any physical address. The source node may perform the translation, or it may leave the logical address untranslated in the message. In the latter case, each intermediate node performs the translation (without changing the logical address in the packet header) before routing the message on the next line; this may result in slightly better route selection. On the other hand, it does not allow subscribers to refer to logical addresses as a part of a group address (as explained in Section IV). In this paper we will assume the source node performs the translation to permit the delivery mechanism for group addressing proposed in Section IV.

Logical addressing also permits multiple homing of subscribers to network ports and the use of one network port for the connection of several distinct subscribers. It is necessary for the source subscriber to identify itself by means of its logical address in the message header, as well as stipulating the destination logical address, if a completely general mapping is desirable.

Physical addressing represents one end of the spectrum of message addressing. One difficulty with this approach is that changes in physical addresses must be announced to all subscribers, with the inevitable operational problems such cha

owing in the net-
iability (to mini-
lures):
at some networks
al addressing sys-
the network, and
gions, may result
s not need to dial
other operational
used in combina-
cuss in this paper.
l to many kinds of
ses on the example
rcuit networks (in
nections analogous
ry efficient logical
essing information
on. On the other
message is handled
e less efficient for
oadcast and group
cessary to set up
fact, it may be
ination addressing
ne extensive control
ircuit networks will
only one of several
uits and datagrams
o methods has been
example). We hope
to that discussion

TIPLE HOMING

can translate many
ess and one physical
tual circuit network
urce node once per
given virtual circuit
In a datagram net-
f messages are trans-
to any physical ad-
translation, or it may
the message. In the
orms the translation
the packet header)
line; this may result
other hand, it does
ldresses as a part of a
). In this paper, we
translation to permit
sing proposed in Sec-

iple homing of sub-
one network port for
ibers. It is necessary
tself by means of its
well as stipulating the
ly general mapping is

d of the spectrum
this approach is the
announced to all sub-
problems such changes

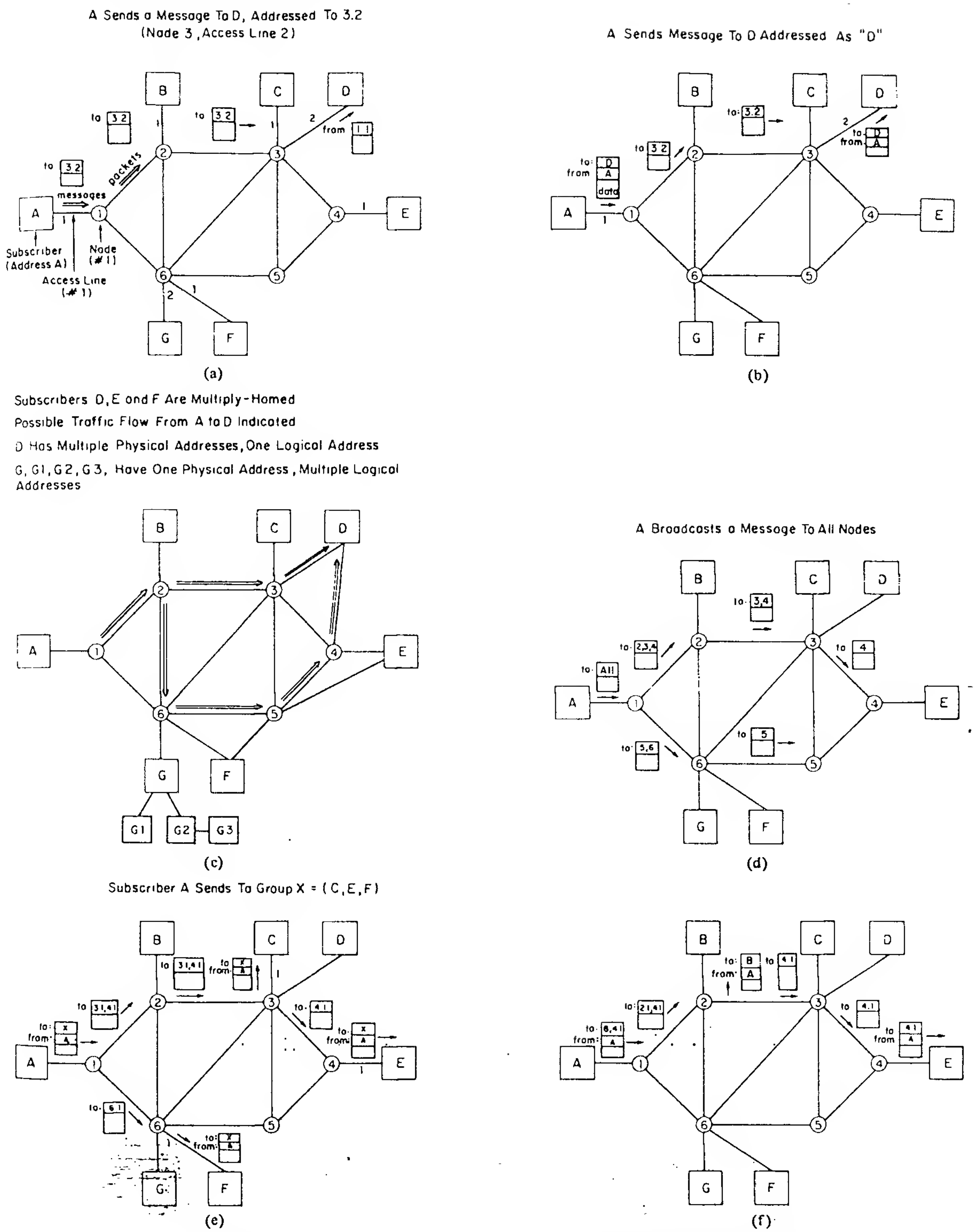

Fig. 1. Message addressing modes—terminology. (a) Physical addressing. (b) Logical addressing. (c) Multiple homing. (d) Broadcast addressing. (e) Group addressing. (f) Multidestination addressing.

...al. Logical addressing for every subscriber is at the other end of the spectrum. In this case, the communications network has the responsibility for keeping track of the location of each subscriber and translating the logical addresses used by subscribers into physical addresses for internal data routing. It is also possible to design hybrid approaches which intermix physical and logical addressing.

A. Implementation Considerations

Logical addressing of subscribers requires some form of "mapping table" for translation between logical and physical addresses. These tables must be stored at one or more locations in the network and updated when changes occur. The cost of this maintenance depends on the size of the network

and the implementation of logical addressing selected. A number of possible implementations are considered below and their costs compared. We select among several possible locations for the mapping tables: partial tables at each node, complete tables at each node, a distributed data base of tables at the nodes, and a centralized table at one or several locations.

1) Physical and Logical Addressing: This is a hybrid approach which may be useful when a network designed for physical addressing only is modified to permit logical addressing. There may be a transition period during which subscribers may use either method, and there may be a requirement to keep both methods if certain subscribers do not implement logical addressing. A subscriber uses either a physical or a logical address for each message which it transmits to its node, and identifies the type of address transmitted with an indicator in the message header. Logical-to-physical address mapping is performed at the source node for certain subscribers. The mapping table consists of N entries giving node number and port number, where N is the number of logically addressable subscribers in the entire network. When N is relatively small, the cost of the table in terms of storage required is insignificant.

One shortcoming of this hybrid approach is that it does not solve the problem of physically addressed subscribers moving from one port to another. Since many subscribers have to change ports or nodes from time to time, there may be considerable operational difficulty in keeping all subscribers informed about physical addresses (a "telephone book" may have to be published regularly). The next two paragraphs suggest techniques for permitting all subscribers to use the logical addressing capability.

a) Logical addressing–Complete mapping: The complete mapping approach to providing a logical addressing capability for subscribers extends the ideas above to include all subscribers. The mapping table structure is the same, though its size is considerably larger since there is one entry for each of the subscribers. Even so, the table may not be impractical to store in primary memory for up to several thousand subscribers.

b) Logical addressing–Partitioned mapping: A different structure for the address table can be developed by taking advantage of the fact that, for routing purposes, a source node needs only the node information in the physical address, and the destination node needs only the port information. Routing is naturally partitioned into two stages; the mapping process can be partitioned in a similar fashion. The mapping table at node X can be divided into two tables, the first with K entries containing node numbers, the second with M entries containing port numbers, where K is the total number of logical addresses except those addresses of subscribers connected to node X, and M is the total number of logical addresses of subscribers connected to node X. Multiply homed subscribers would be associated with one node at a time. The actual implementation of these two logical tables could be a single table with an entry for every logical address containing a data field and a Boolean variable to distinguish node entries from port entries.

c) Logical addressing–Information service: This approach is based on the existence of one or more information service centers on the network. The center(s) would maintain the address mapping information for the network subscribers and provide it to the nodes upon demand. Under IBM's SNA, the System Services Control Point (SSCP) provides such a function. This approach is probably most useful for large networks in which there are a few large central nodes and many smaller nodes with reduced capabilities. Each smaller node might maintain a set of address transformations used recently, together with those for its active connections and perhaps some others, to avoid access to the service centers for every message.

2) Relation to Multiple Homing: In many cases it is desirable to connect subscribers to more than one network node to improve reliability (and also to provide additional bandwidth over the multiple paths if they can be used simultaneously). Several connections to the same node can also be used. Any of the logical addressing techniques can be used to support multiple homing, provided that multiple entries are present in the address translation table. There are three approaches to routing messages over multiple access lines. The simplest approach is to use only one at a time. For datagram networks, it is possible to route each message to the "best" access line, e.g., the one which minimizes delay. For virtual circuit networks, an alternative approach is to route entire virtual calls to one access line or the other, independently selecting the access line for each virtual circuit. For each approach it is useful to consider the issues of efficiency, reliability, and switchover management, routing (access line selection), and message processing (network error control, flow control, and sequencing); these topics are covered in the next two subsections.

B. Efficiency Considerations

For a virtual circuit network, logical addressing can be implemented by exchanging the appropriate mapping information between the source and destination nodes as part of the connection setup procedure. The result of this exchange is that the source and destination nodes each remember the physical address and logical address of the subscriber at the other end. They can be used without reference to the address mapping table for the duration of the logical connection. This is an efficiency advantage not shared by datagram networks. Specifically, in a virtual circuit net the packets flowing in the network can be addressed with the physical address of the destination subscriber only, and the message header for the destination subscriber can be constructed at the destination node. This message header must contain the logical address of both source and destination subscribers. In a datagram net the packet header must contain both the logical address information for the subscribers and the physical address information for network routing. Since the main address translation table is referenced only at call setup time in the virtual circuit case, it may be practical to store the logical-to-physical mapping table on secondary storage if available. Thus it appears that virtual circuits, once established, are more efficient for logical addressing than datagrams.

With respect to maintaining the address mapping table, the alternatives are central versus distributed and automatic (adaptive) versus manual updating. A distributed adaptive approach, similar in concept to the ARPANET routing algorithm [1], is attractive. In this method, each node is responsible for the subscribers connected to it. When the set of subscribers connected to it changes, it attempts to send this information (automatically) to the other switches in the network.

C. Reliability Considerations

The key difference between the network processing for multiply homed subscribers and for singly homed subscribers is that procedures must be defined for switching logical con-

tions from one line to another in case of a failure. Four cases must be distinguished. The dual-homed subscriber may be the source or destination. In each case, the failure that necessitates switchover may occur in the node or on the access line. These four cases are explained below, where we define how failure is detected and what action is taken to reestablish the existing logical connections via the alternate line.

a) Case 1–Source Node Fails: The source subscriber can learn that switchover is required by observing the flow of actual or artificial traffic exchanged over its access line. It may have imperfect information concerning the disposition of the messages it has sent into the network. The subscriber level protocol should have sufficient error-control capability so that missing or duplicate messages can be detected.

b) Case 2–Source Access Line Fails: This case could be treated exactly like Case 1: however, since state information on all interrupted logical connections is still available at the source node, a better plan of action is possible. A special control mechanism can be added to the network to permit the source node to forward all of its state information on logical channels associated with the source subscriber to another node to which the subscriber is connected. The subscriber could then continue merely by retransmitting any message that was unacknowledged when the access circuit failed. Even though the subscriber level protocol must be prepared to deal with Case 1, it may be more efficient to deal with Case 2 at the network level, making it invisible to the subscriber.

c) Case 3–Destination Node Fails or Is Inaccessible: The source node will learn that the destination node is unavailable via the network routing information. The source node can decide to use the alternate address for the destination subscriber and route to other destination nodes. If copies of the messages are kept at the source node, either the source node or the source subscriber can initiate retransmission. The subscriber protocol must then be prepared to handle duplicates.

d) Case 4–Destination Access Line Fails: The source subscriber can learn of the problem via a "Destination Down" message in response to one of its data messages. The source node can decide to use the backup destination access line; connection resynchronization is similar to Case 3 except that additional state information about each logical channel is available at the destination node. This information may be sent to the source node (or possibly to the new destination node, but that would be more complex) in order to return to the source subscriber all acknowledgments queued at the destination node. Message duplication is still a possibility and either the source node itself or the source subscriber might actually effect the retransmission.

D. Datagram Considerations

Routing each message independently to one access line or the other satisfies the reliability objective and is attractive in terms of the goal of providing flexible allocation of access line bandwidth. However, there are costs associated with the more complex routing and message processing required.

1) Routing: Each message is independently routed to one of the access lines connecting the destination subscriber to the network. The source subscriber or the source node can direct the message to its destination; however, in neither case does the source entity have information about the present or future traffic over the access lines to the destination subscriber. To eliminate these problems the network routing algorithm may be designed to incorporate routing information concerning multiply homed subscribers so that each node knows its best route to each multiply homed subscriber. In other words, if a subscriber has more than one access line, and if any message can flow over any access line, then the access line selection can be treated as a routing problem rather than an addressing problem. The routing process must deal with choosing routes to nodes with more than one line, so it can be augmented to deal with multiply homed subscribers as well.

2) Message Processing: By message processing we mean error control, flow control, and sequencing. Dynamically assigning datagrams to access lines requires dual-homed subscribers to do message processing themselves. Any attempt to use multiple logical channels for a single logical data stream requires the destination subscriber to be involved in reordering and related functions. An error control mechanism is required to handle both missing and duplicate messages. Reordering at the destination subscriber is required if either the destination subscriber or the source subscriber is multiply homed; sequence numbers can be attached to messages by the source subscriber to allow the messages to be identified and reordered by the destination subscriber.

3) Switchover Management: Detection of a failure and switchover are as described in Section II-C. Since an alternate logical connection from the source to the destination already exists, the source subscriber needs to retransmit only those messages whose disposition is unknown at the time of the failure.

E. Virtual Circuit Considerations

Associating all of the messages of a virtual circuit or "conversation" with a single access circuit is a compromise which not only provides high reliability and makes relatively efficient use of the existing access circuit bandwidth, but also is well suited to the logical addressing schemes described in Section II-A.

1) Routing: If a multiply homed subscriber is connected to different nodes, the source node establishes a logical channel for the entire conversation to a destination node selected from among alternatives in its logical-to-physical address mapping table, based on current routing data. If a subscriber is multiply homed to a single node, only a single entry exists in the mapping table at the source, and access circuit selection occurs at the destination node. The destination node records multiple port numbers for each of its multiply homed subscribers in its address mapping table and selects one when the "call request" message associated with the conversation is received. The structure of the address mapping table must permit multiple nodes and ports for a given address in order to implement this scheme.

2) Message Processing and Switchover: As in Section II-C, no message processing functions are required of the subscribers. The method of detecting that logical channels need to be switched from one access line to another and the procedure for effecting the resynchronization are also identical to the methods and procedures described above.

III. Broadcast Addressing

Broadcast addressing means the capability for one node to send a message to all other nodes by marking it with the address "ALL" rather than by sending separate messages to each node. This topic has not yet received much attention. Dalal's 1977 dissertation, "Broadcast protocols in packet-switched computer networks" [3], discusses the design and analysis of

broadcast routing algorithms for use in packet-switched computer networks. Five alternatives are considered in terms of qualitative implementation and quantitative performance. Many internal network algorithms as well as subscriber applications require an identical data base at all nodes. For instance, the logical addressing algorithms discussed above assumed an identical address translation table at each node. Also, as explained below, some adaptive routing algorithms depend on having up-to-date information on all network topology and traffic. Broadcast addressing can be used for the propagation of information throughout the network to all nodes. For this reason, we will sometimes refer to the broadcast messages as "updates" in the discussion below.

To provide additional background for this topic, it is useful to consider the changes proposed for the ARPANET routing algorithm. The current ARPANET routing algorithm [1] determines which nodes are reachable from a given node (by exchanging routing update messages with other nodes) within several seconds of a change. We would like to shorten this time, since we have observed, on occasion, that congestion can build even in these few seconds [4]. The basic reason for the delay in adaptation rests with using a routing algorithm which has information on entire paths only, not individual lines, and which relies on hop counts and timers to determine whether nodes are reachable. Since this is an essential feature of any ARPANET-like algorithm, we were led to consider other types of procedures to increase the speed of adaptation, while reducing the cost.

It is practical to implement a separate and independent shortest path calculation in each of the IMP's in the ARPANET as opposed to the present distributed computation [5]. Such an algorithm can be designed to be very efficient in space and time, using as little as 1 or 2 ms of CPU time, on the average, to perform an individual update when the calculation is performed incrementally. Efficient and reliable updating procedures can be developed so that a shortest path algorithm can be performed on an event-driven basis.

The shortest path algorithm has significant advantages over the present ARPANET algorithm in terms of efficiency, reliability, loop freedom, and speed of adaptation. The basic algorithm can be attributed to Dijkstra [6]; because of its search rule, we call it the shortest path first (SPF) algorithm. The algorithm used to generate the shortest path tree initially is illustrated in Fig. 2. (We have also developed an algorithm for modifying the tree incrementally when network changes occur.)

The basic algorithm for finding the shortest path tree from a given source node ~~is a way~~ of building up the tree node by node. That is, the ~~tree initially~~ consists of just the source node. Then the tree is augmented to contain the node that is closest to the source and that is adjacent to a node already on the tree. The process continues by repetition of this last step. The tree is built up SPF–hence the name of the algorithm. Eventually, the furthest node from the source is added to the tree, and the algorithm terminates. As a by-product of the algorithm, it is simple to produce a routing directory. In the ARPANET, each node will run the algorithm with itself as the source. In order to run the algorithm, each node must maintain a data base representing the topology of the network. A key component in the data base is the "length" of every line in the network (where "length" is not physical length, but rather some relevant metric such as delay). This data base can be updated using broadcast addressing to distribute information from the node which discovers a change on one of its lines (e.g., in delay or error rate) to all other network nodes.

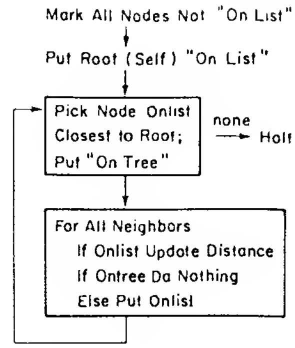

Fig. 2. Shortest path first routing algorithm (SPF).

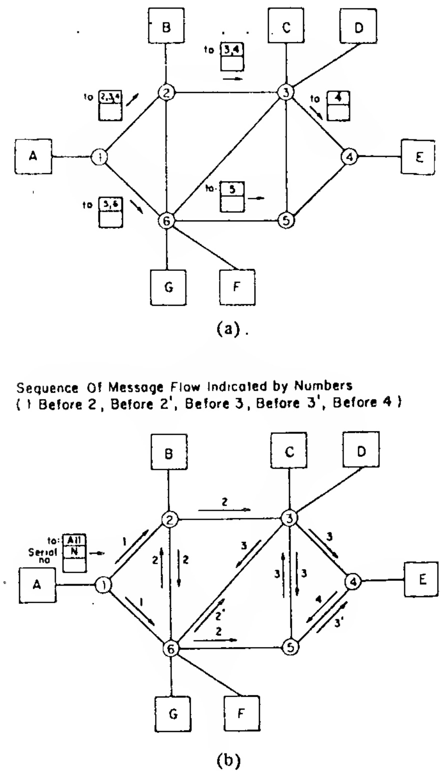

Fig. 3. Broadcasting and flooding. (a) Broadcasting. (b) Flooding.

A. Implementation Considerations

There are two general types of approaches to the problem of transmitting a message to all possible addresses: to route it once to each node (termed "broadcasting" here), or to send a copy over each network line (termed "flooding"). Flooding may be simpler to implement, but the nodes receive multiple copies of the message.

Broadcasting is a system in which the source explicitly addresses the message to all nodes, by labeling one or more copies with the appropriate addresses for each of its output lines, and routing them to each node along the best path (see Fig. 3(a)). Such a scheme can require as little as N-1 packet hops for the broadcast, which is optimal (N is the number of nodes in the

twork). Each broadcast address is represented as an N-bit ctor, each bit indicating whether the message should be sent the corresponding node. The N bits are needed to indicate hich nodes have received the message so far and which nodes ve not; bits are turned off as the message flows through the twork. The source of the message sets the address of the ssage transmitted on each of its lines to have bits corre-nding to those nodes for which that line is the best route. her nodes receiving such messages turn off their bit and then rform the following operation on the resulting address: for h adjacent node, take the logical AND of the received ad-ss and the bit vector of nodes for which the adjacent node he best route. If nonzero, send a message with the resulting ND'ed address to that adjacent node. Broadcasting is the neral method for sending a message to multiple destinations, d will be referred to again in Section IV. However, in the cial case of addressing all nodes, the next method may be ferable.

Flooding is a method in which each node sends each "new" date on all its lines except the line on which the update was eived (see Fig. 3(b)). A new update is one the node has not n before; the serial number is larger than the last one re-ed for that particular node. This requires $L - N + 1$ packet s (where L is the number of lines in the net, counting each ection separately), since an update will flow on all lines ex-t "backwards" on the $N - 1$ lines of the broadcast tree from e source. If we define $L = cN$, where c is the average node nectivity, then this number of updates is $cN - (N - 1) =$ $1)N + 1$.

e assume that the technique for addressing all nodes must t the following criteria:

	Normal Operations	Node Failure or Partition	Node Recovery or Partition End
iency	low CPU and line overhead	fast notification at low overhead	
bility	sequencing of multiple updates	no loss of updates	complete information made available

urally, it is difficult to meet all of these objectives; effi-cy and reliability are discussed in turn below.

fficiency Considerations

e important consideration in efficiency is line bandwidth; bandwidth requirements can be shown to be very small:

Broadcasting: The message is $b + N$ bits long, where b is number of bits in the body and N is the N-bit address. s the total number of bits on all lines is $(b + N)(N - 1)$. refore, the total bit rate per line if each node updates every onds is

$$\frac{(b + N)(N - 1)}{ct}.$$

Flooding: The message length is only b bits, so that the al number of bits on all lines is $b((c - 1)N + 1)$. The total ate per line is

$$\frac{b((c - 1)N + 1)}{ct}.$$

adcasting is quadratic in N, which means for large enough will become more expensive than flooding. The crossover

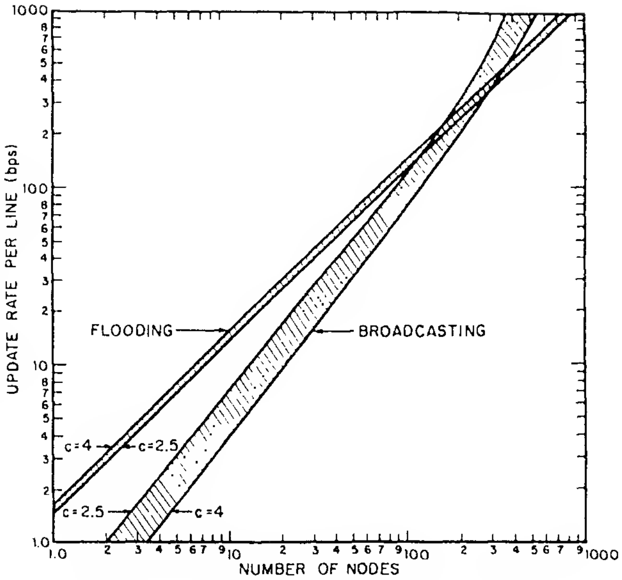

Fig. 4. Overhead per line (assuming 1 update/node/100 s).

point occurs when

$$(b + N)(N - 1) = b((c - 1)N + 1).$$

This is illustrated in Fig. 4 below for $b = 200$ bits. $t = 100$ s. For $c = 2.5$, crossover comes near $(200) \times (0.5) = 100$ nodes. For $c = 4$, crossover is not until 400 nodes. Note, however, that for $c = 2.5$ both methods require less than 100 bits/s (0.2 percent of 50 kbits) to update 80 nodes at a rate of once every 100 s. The line overhead scales linearly with t, the up-date rate, so other strategies can be compared simply by re-labelling the γ axis. (For instance, if $t = 10$ s, then updating in the ARPANET with $N = 62$, $c = 2.5$ would require 750 bits/s which is only 1.5 percent of 50 kbits).

Some interesting points emerge from examination of Fig. 4:

1) The overhead for flooding can be plotted as a straight line on log–log paper, with practically no dependence on c. This makes it useful for long-range planning, since it is not sen-sitive to network topology.

2) For a given number of nodes, flooding grows less efficient as the net is more highly connected, while broadcasting grows more efficient.

3) The two methods are quite similar for networks with 50 to 200 nodes.

4) The magnitude of the updating overhead is very low, even for large nets.

Since flooding is more efficient for large nets, and is very efficient in absolute terms for small nets, it is the best overall choice for efficiency.

A second important advantage of flooding is that the node sends the same message on all its lines, as opposed to creating separate messages with different bit-vector addresses on the different lines. This may make it considerably simpler to program the broadcast addressing mechanism, since there is no problem of reserving buffers or dealing with the situation in which the node has no more buffers for copies of messages. A final consideration which favors flooding is that it does not depend on the correct operation of the routing algorithm, and is therefore less sensitive to network failures. This makes it a safer, more reliable system than broadcasting.

C. Reliability Considerations

At first glance, it seems essential to acknowledge (ACK) messages to make sure they get through. This is useful for flooding; if messages are acknowledged at each hop, then with flooding the message will be received at all nodes which have a path to the source at the time of the transmission. On the other hand, with broadcasting using an n-bit vector, transmission is not reliable. Two examples are 1) a node which receives an update, acks it, and then fails, and 2) a section of the network which is partitioned from the rest while an update is flowing through it destined for the main body of the net. In each case, an update will be lost. Under broadcasting, acknowledging updates at each hop is not sufficient to ensure reliable updating of all nodes which have a path to the source at the time of the update. If a positive acknowledge/retransmission system is used, then appropriate data and control structures are needed in the node. Some possibilities are:

Method	Problems
Separate ACK's	
Invent a new set of logical channels for routing, common to all lines.	adds complexity to the node (packets on multiple queues, etc.)
Periodic ACK's	
Send all ACK's periodically rather than one at a time in separate messages.	slower reaction to a lost update than usual ACK system.
One last possibility, though not an ack system, is:	
Periodic rebroadcast	
Send the update once only, and rely on a periodic retransmission to ensure it gets through.	even slower error recovery, though very reliable in the long run.

If separate ACK's are used, the ACK could look very similar to the packet acknowledgments. The expected number of updates per second would be very small, since there are very few updates/line/second with either broadcasting or flooding. (For instance, if updates were generated once a minute on the average by each ARPANET node, then an update message would flow over each line every 2 s on the average.)

For periodic ACK's, one could use the periodic test messages exchanged between nodes (termed "Hellos" here) to carry a 1-bit ACK for each node. This would require an additional N bits per Hello message, rather than a separate ack message for each update. A drawback of this scheme is that the sender is "blocked" from sending another update about some node until the previous one is ACKed.

We now compare these two possibilities in terms of the extra line overhead they require. In both cases, we will assume a 136-bit Hello message (the same length as a separate ack message) is sent every 640 ms, contributing 212 bits/s of overhead, and we are concerned only with additional overhead beyond this. The amount of line overhead used by the two ack methods is shown in Fig. 5, for both broadcasting and flooding in the case of separate acks, for $t = 10$ s, and $t = 100$ s. The following statements are true for all values of N (all network sizes):

Separate ACK's with flooding use 50 percent more bandwidth than separate ACK's with broadcasting.

All ACK's in the Hello are better than separate ACK's if t, the routing update period, is small.

Any ACK method will contribute a significant percentage to line overhead, as much as doubling the routing overhead, but this is probably a small factor in absolute terms.

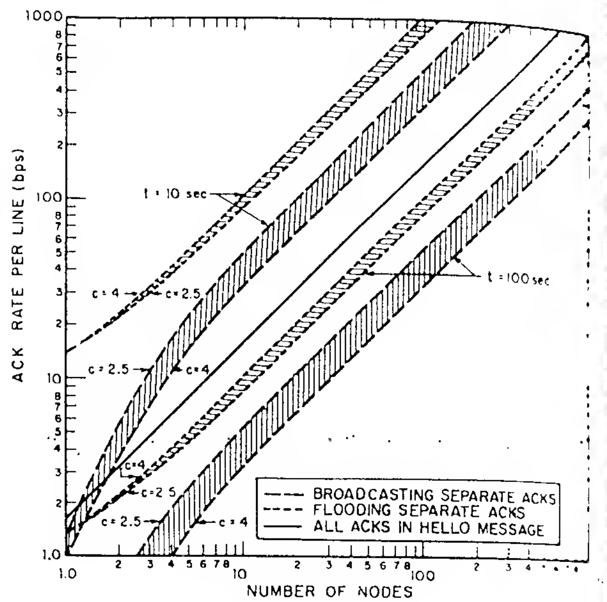

Fig. 5. Acknowledgment overhead per line.

The total overhead associated with routing is the sum of the routing update, acknowledgment, and Hello overheads. For the cases under consideration, we have

Broadcasting:

$$\text{Separate ACK} \qquad \frac{(336+N)(N-1)}{ct} + \frac{136}{64}$$

$$\text{ACK in Hello} \qquad \frac{(200+N)(N-1)}{ct} + \frac{136+N}{64}$$

Flooding:

$$\text{Separate ACK} \qquad \frac{336((c-1)N+1)}{ct} + \frac{136}{64}$$

$$\text{ACK in Hello} \qquad \frac{200((c-1)N+1)}{ct} + \frac{136+N}{64}$$

These four cases are compared in Fig. 6 for $t = 100$ s. It is obvious that the line overhead is dominated by the Hello messages for networks with less than 100 nodes. In this case it is possible to choose among the updating alternatives presented here on grounds other than line overhead.

As an additional precaution, especially during test and installation, it is possible to reflood the net periodically with the update information from each source. For instance, the periodic rate could be set at 1/100 s, which would add only [illegible] bits/s to each network line.

With any ACK method, there are other difficulties to be solved. With a separate ACK scheme, the sender must keep a timer for each un-ACKed update, and resend it periodically. With the ACK's carried in Hello messages, there is the chance that the receiver will have just sent a Hello containing the old serial number bit when it receives the new update, causing the sender to retransmit the update unnecessarily. The probability of the update and an "old" ACK crossing in mid-flight is

$$\frac{s+r}{p}$$

where s is the time to send update, r is the time to send Hello

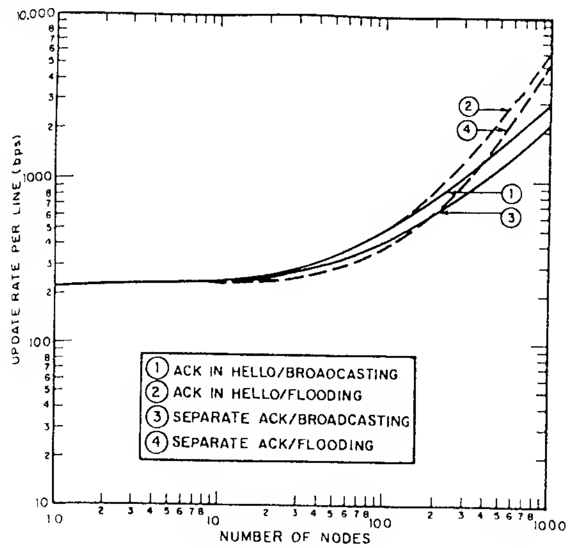

Fig. 6. Total overhead per line.

.K, and p is the period for sending Hellos (0.64 s). For ...pical land lines s and r will be equal and small (about 25 ms ...h). Thus

$$\frac{s+r}{p} = \frac{5\text{ ms}}{640\text{ ms}} = 8 \text{ percent spurious retransmissions.}$$

...lternatively, the node could keep a clock for each destina-...n which would ignore any Hello/ACK's within the last x ...lliseconds. This would be necessary on satellite lines, where ...: probability of retransmission without the timer becomes ...se to 1. There the timer must be longer, e.g., 600 ms.

...hen a node comes up, or when it returns from a partitioned ...e (in which it was isolated from several other network nodes ...pically when its line(s) to the network were down) it must ...complete update of all information contained in broadcast ...lates, since an indeterminate number of updates have taken ...e. Two possibilities exist:

...) The two adjacent nodes which were isolated from each ...er send each other their entire update tables.

...) The two adjacent nodes exchange table entries for all ...chable nodes.

...he first is somewhat simpler to program, but uses more ... bandwidth than the second. If the table is garbage-...ected or compacted to remove entries for unreachable ...des, then the two methods are identical. If compaction ...ot too difficult, it is probably the best method.

...hen a node receives such an update, possibly containing ...rmation on many nodes previously unreachable, it treats ...ke a normal single-node update, and sends it to its neighbor ...des by the flooding method. This works well for both sides ...en a node that was down comes up; it gets all the tables ...m its neighbor, and the rest of the net gets its own single ...e entry. (Several messages may be required to send all the ...ate information to a node or nodes which were previously ...ated.)

IV. Group Addressing and Multidestination Addressing

...r reasons of convenience and efficiency, it is desirable to ...vide a facility for addressing messages with the name of a ...up of addresses (logical and physical addresses, singly homed

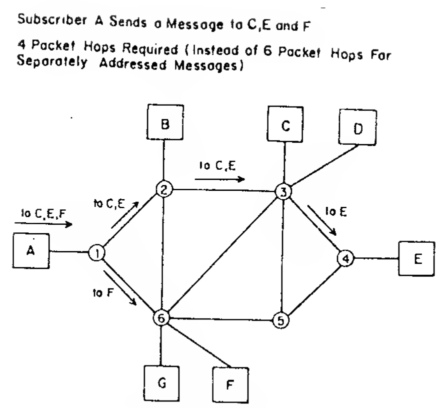

Fig. 7. Group addressing and multidestination addressing.

and multiply homed subscribers). Such a group may correspond to an on-going conference call or distributed working group of some kind, or it may be a simple distribution list for certain messages. In addition to such pre-established group addresses, it may also be useful to provide a general capability for addressing messages to a list of subscribers. This multidestination addressing can cut down on network traffic and subscriber overhead by substituting a single transmission for several separately addressed messages (see Fig. 7).

A. Implementation Considerations

In a virtual circuit net, group addressing and multidestination addressing are unwieldy: both inefficient and difficult to control. The two basic alternatives are to set up $(a) \times (a)$ virtual circuits when a addresses are present in the group, or to modify the packet header to permit multiple message numbers, acknowledgments, and allocations to flow over the same multidestination virtual circuit. Both methods appear to be so complex that it is difficult to justify their implementation. For a datagram with many addresses the problem is simply to route the datagram efficiently to the destinations.

The issues of formatting packets and messages with logical addresses and multidestination addresses deserve some consideration. Group addressing is simpler to implement in the network since it requires a relatively small change to the subscriber software—the group address replaces the usual physical or logical address. On the other hand, multidestination addressing is more flexible and useful to the subscribers but requires a fairly major change in the subscriber-to-network format, since a new variable-length address format is needed. Careful attention must also be given to the interaction between logical addressing and group addressing, since group addressing, in general, should permit reference to logical as well as physical addresses. As an example of this interaction, if a group address refers to several logical addresses as well as physical addresses, then the translation of logical-to-physical addresses must take place at the source node.

B. Efficiency Considerations

The efficiency of a multidestination or group addressing system depends critically on the routing algorithm used. One useful metric for determining the efficiency of a multidestination system is the number of packet hops required to transmit a given packet to all the destinations (the number of hops traversed by each packet summed over all packets transmitted). A simple routing algorithm can be designed for multidestination

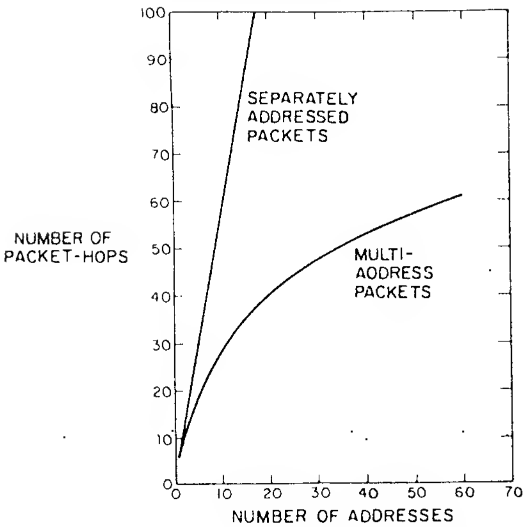

Fig. 8. Multiaddress packets: Number of packet hops.

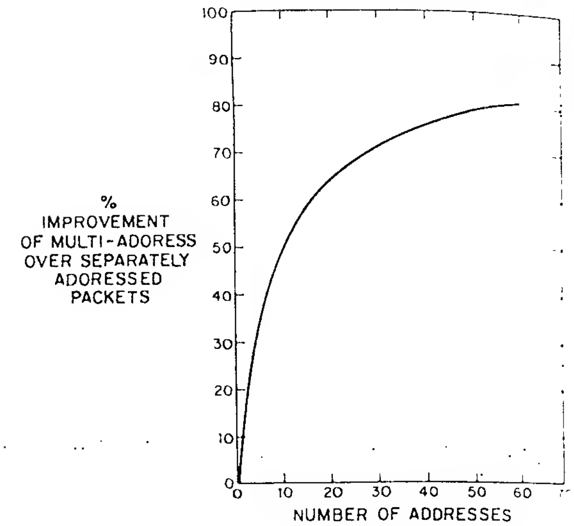

Fig. 9. Multiaddress packets: Percentage improvement.

transmission which is similar to the "broadcasting" technique described in Section III-A. It can be constructed on the basis of a standard routing algorithm for single destination packets (e.g., based on minimum path length or minimum delay). In addition to this algorithm, it is necessary to provide a multidestination address (an N-bit vector indicating the destination nodes) in the header of the packet. When the packet arrives at an intermediate node, the node simply creates as many copies of the packet as there are different routes in the routing directory for the different destinations in the header. Each time multiple copies of a packet are created at a node, each copy is assigned the appropriate subset of the destinations for which that line is the first line on the best path. In this way a broadcast of a given packet to all N-1 other nodes in the network can be accomplished with only N-1 packet hops, which is optimal. (Note that this routing algorithm is not optimal for a message addressed to fewer than N-1 other nodes; some form of minimum spanning tree algorithm is needed to achieve the minimal number of packet hops in the general case).

It is revealing to analyze the performance improvement, measured in packet hops, gained by multiaddress messages using the simple routing procedure based on minimum path length described above. The following definitions will be useful:

N	number of nodes in net,
a	number of addresses,
$p(a)$	number of packet hops,
h	average path length,
c	average node connectivity.

For separately addressed packets, an average of $h \times a$ packet hops are required to transmit a packets. For multiaddress, $p(a)$ is a more complicated function. Clearly, $p(1) = h$ and $p(N-1) = N-1$. A little thought shows that $p(a) < h \times a$ for all a, and $p(a) > a$ for all a. Furthermore, $p(a+b) < p(a) + p(b)$; $p(a)$ is concave downward.

Figs. 8 and 9 show some empirical investigations we made to determine the behavior of $p(a)$ for the ARPANET. Although we have not found a closed-form expression for $p(a)$, a close fit for $p(a)$, based on data from the ARPANET and other networks, is:

$$p(a) = ha - (h-1)a \times \log_{N-1}(a).$$

That is, the percentage improvement of multiaddress over separate addresses is given (approximately) by

$$(h-1)a \times \log_{N-1}(a).$$

Thus in the ARPANET ($N = 62$, $c = 2.5$, $h = 5.5$), the greatest savings in number of packet hops, 80 percent, occurs when addressing all other nodes in the network. When addressing eight destinations (equal to the square root of the number of nodes in the network), half of this relative improvement, or about 40 percent, is obtained. We have calculated that addressing as few as 5 to 10 destinations in the same packet results in a savings of 25 to 50 percent of the packet hops required in the ARPANET with separately addressed packets

C. Reliability Considerations

Since we have assumed that group addressing and multidestination addressing are not practical for virtual circuit service and should be implemented only for datagrams, the subscriber using these services must take responsibility for providing reliable transmission for the end users. A host-level protocol is necessary between the sending subscriber and each receiver to ensure an error-free, sequenced flow of messages between each pair of subscribers.

V. Conclusions

The enhanced message addressing modes discussed in this paper have several important advantages over physical addressing. Logical addressing provides for considerable operational flexibility and reliability. The use of multidestination and group addressing has been shown to lead to significant reduction in network traffic, even for the case of relatively few destinations per message. One of the important conclusions from this work is that while virtual circuit networks have some efficiency advantages over datagram networks for logical addressing, datagram networks facilitate the use of broadcast addressing and multidestination addressing. These important points of comparison have not yet been fully considered by the network design community.

While this paper has focused almost exclusively on the example of terrestrial packet switching networks, many of the results

can be extended to other kinds of networks. Of particular interest is the use of broadcast and group addressing in networks which contain broadcast/multiple-access links, such as satellite and radio channels.

ACKNOWLEDGMENT

The ideas reported on in this paper were developed in discussion with several people, especially R. E. Kahn and V. G. Cerf of ARPA and G. Falk, I. Richer, and E. C. Rosen of BBN. R. E. Kahn contributed many helpful suggestions to the presentation and content of the paper, for which the author is very grateful.

REFERENCES

[1] J. M. McQuillan and D. C. Walden, "The ARPANET design decisions," *Comp. Net.*, vol. 1, no. 5, Sept. 1977.
[2] L. G. Roberts, "The evolution of packet switching," this issue, pp. 1307-1313.
[3] Y. K. Dalal, "Broadcast protocols in packet-switched computer networks," Ph.D. dissertation, Stanford University, Digital Systems Laboratory Tech. Rep. 128, Apr. 1977.
[4] J. M. McQuillan, G. Falk, and I. Richer, "A review of the development and performance of the ARPANET routing algorithm," submitted to *IEEE Trans. Commun.*
[5] J. M. McQuillan, I. Richer, and E. C. Rosen, "ARPANET routing algorithm improvements–First semiannual technical report," BBN Rep. 3803, Apr. 1978.
[6] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numer. Math.*, vol. 1, pp. 269-271, 1959.

Commercial, Legal, and International Aspects of Packet Communications

STUART L. MATHISON, MEMBER, IEEE

Invited Paper

Abstract—Packet switching technology emerged rapidly in the 1970's as another viable mode of communications switching, along with circuit and message switching. Since packet switching offers economical and versatile data communication capabilities in a multiuser environment, it is particularly well suited for furnishing public data communication network services.

Public packet networks are now established or being developed in most industrialized countries, and the introduction of these networks has raised policy issues relating to the structure and regulation of the national networks, and the interconnection of national networks into an international packet switching system.

This paper reviews these issues and concludes that public packet switching network services will continue to be regulated in all cases; that competitive packet networks will coexist in the U.S. and in Canada, but that only one national packet network will exist in each of most other countries; that packet networks will aggravate the problem of distinguishing nonregulated data processing services from regulated data communication services; that international interconnection of public packet networks based upon CCITT standards will occur rapidly over the next several years; and that a unified international packet switching system will eventually emerge similar to today's international telephone and telex systems.

Manuscript received February 21, 1978; revised July 22, 1978.
The author is with the Telenet Communications Corporation, Washington, DC 20036.
Note: The opinions expressed in this paper are those of the author and should not be taken to reflect the views of Telenet Communications Corporation.

I. INTRODUCTION

DURING the mid-1970's public packet switching networks emerged in a number of countries, offering computer users and communicators a highly cost-effective and versatile means of transferring digital data between terminals and computers. The introduction of these new services has raised several national and international policy issues, the resolution of which will affect computer users, computer equipment manufacturers, communication common carriers, and the general communicating public.

This paper is intended to introduce the reader to these policy issues, to review the principal arguments expressed on each issue, and to indicate the likely policy resolution in the future. The paper is organized into two major sections, the first covering the structure and regulation of packet switching services at the national level, and the second section covering the structure, pricing, and standards-making activities relating to packet networks at the international level.

A typical packet switching network consists of many distributed store-and-forward switching centers, multiply interconnected. Such a network is similar to many private data communication networks in that it may be implemented by leasing the communication channels from the traditional communica-

PROCEEDINGS OF THE IEEE

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

NOVEMBER 1978

SPECIAL ISSUE ON packet communication networks